社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2004－HPC－99 (21)
2004／7／31

10Gb Ethernetを用いた高性能通信機構の設計

住元真司† 佐藤充† 中島耕太†
久門耕一† 石川裕††

本論文では、10Gbps ネットワークを用いた高信頼高性能システム 向けの通信機構の設計について述べる。現状のハードウェアで 10Gbps の通信性能を生かし切るには、RDMA 通信を用いる必要がある。しかし、現状の RDMA 通信モデルはページアウトを避けるため Pin-down を実行する。このためネットワークを用いた同期を行う必要がある。そこで本稿では、ネットワーク経由での同期を不要として、低コストで高い通信性能を実現する RDMA 処理方式の設計について議論する。

A Design of High Performance Communication Facility Using 10Gigabit Ethernet

SHINJI SUMIMOTO,† MITSURU SATO,† KOUTA NAKASHIMA,†
KOUICHI KUMON† and YUTAKA ISHIKAWA ††

This paper designs a high performance communication communication facility using 10Gb Ethernet. RDMA Communication is needed to achieve higher communication bandwidth performance using 10Gbps class network. However, existing RDMA methods reqiures pin-down to avoid page-out, and synchronization between nodes using network. Therefore, this paper discusses low-overhead methods which do not require the synchronization.

1. はじめに

近年、PC や Ethernet ネットワークなどコモディティハードウェアの高性能化と低価格化が進んでいる。PC のプロセッサの動作周波数は既に 3GHz を越え、コモディティネットワークである Ethernet についても、既に 100MB/s 以上のデータ転送能力を持つ Gigabit Ethernet が普及している。コモディティネットワークの性能の向上は目覚しく、次世代の 1.25GB/s の転送能力を持つ 10Gigabit Ethernet が製品化が始まり、システムバスの転送能力に迫ろうとしている。

このような背景から、我々は、PC とコモディティネットワークを用いたシステム上で、現在共有メモリ型の大型サーバで実行されているデータベースなどのビジネスアプリケーションが実行可能になると考えており、このためのアーキテクチャである次世代高性能アーキテクチャ [1] の研究開発を行っている。

次世代高性能アーキテクチャでは、サーバ間接続として 10Gb Ethernet を採用する。10Gb Ethernet は、1.25GB/s の転送能力を持つが、現在よく利用されている PC ハードウェアと TCP/IP プロトコルの組合せでは十分にハードウェア性能を引き出せていない。TCP/IP のプロトコル処理には文献 [2),3)] に書かれた種々の問題があるが、特に PC ハードウェアのメモリバンド幅が十分でなく、プロセッサによるコピー性能が転送性能のボトルネックとなっている。

プロセッサのコピーボトルネックを回避するために、ネットワークインターフェイス (NIC) から直接 PC のメモリにデータ転送を行う RDMA(Remote Direct Memory Access、または Zero-Copy 通信) と呼ばれる通信方式がある。しかし、現状利用されている RDMA 通信方式は、データ転送前に転送先との間で同期をとる必要があるため、メッセージ長が小さい場合に転送性能が落ちる問題があった。本論文では、この問題を解決し高い通信性能を実現する通信方式について議論する。

2. 次世代高性能アーキテクチャでの高性能通信の要件

次世代高性能アーキテクチャ(図 1) では、サーバ間接続として 10Gb Ethernet を採用する。

次世代高性能アーキテクチャにより実現されるシステムでは、共有メモリシステムと同等の機能をを PC クラスタで実現するために非常に高性能な通信 (高バ

† 富士通研究所
FUJITSU LABORATRIES
†† 東京大学
The Tokyo University

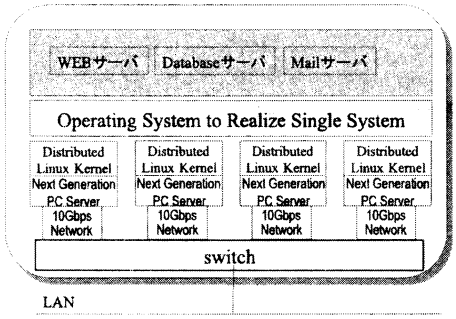

図 1 次世代高性能アーキテクチャ

ンド幅と低遅延) が要求される。通信の用途としては、ファイル I/O(キャッシュ)、遠隔プロセス生成、遠隔プロセススワップ、マイグレーションを想定している。

以上の用途については、OS が管理するページやブロック単位での転送が主体となり、転送単位は数十 KB 単位から GB クラスにまで及ぶ。また、通信時における CPU 占有率は可能な限り小さいことが望ましい。

3. RDMA 通信とその課題

最近の PC のプロセッサとネットワークのハードウェア性能の向上は目覚しく、これらの性能向上に比べてメモリ性能が性能上のボトルネックとなっている [3]。

特に、10Gbps クラスの高速ネットワークのハードウェア性能を最大限に引き出す場合、ホストプロセッサによるデータのコピー性能が通信性能に与える影響は大きく、ホストプロセッサによるコピーを伴わないデータ転送方式の採用が必須となっている。

ホストプロセッサのコピーを伴わないデータ転送方式として、RDMA(Remote Direct Memory Access, aka Zero-Copy 通信) 方式が使われている (以降、RDMA 方式)。この方式はネットワークインターフェイス (NIC) から直接ホストのメモリに送受信データを転送する方式で、転送元と転送先のアドレス (あるいは、それぞれを一意化する ID とオフセット) を指定してデータ転送を行う。転送時には OS に対して転送対象領域の page out(もしくは プロセスの Swap) が発生しないように Pin-down (Mlock) と呼ばれる処理を実行し page out を回避する。

この Pin-down 処理は、強制的に page out を禁止するため、全体メモリ量に対する Pin-down を実施するメモリ量を適切に制御して、OS の動作に悪影響を与えないようにする必要がある。このため、通信機構で Pin-down できるメモリ量を抑制する手法が用いられる。利用するアプリケーションは Pin-down できるメモリ量を意識してプログラムを書く必要があった。

しかし、本方式は、双方のメモリ領域が Pin-down されていることを確認する必要があるため、結局、ネットワークを経由して同期を取る必要がある。これは、最低ラウンドトリップ遅延分のオーバヘッドが加わるため、低通信遅延を必要とするアプリケーションの場合に問題となっている。このため、低遅延を要する場合は、メッセージ通信を用いるなど、RDMA 方式を使わない場合が多い。

このように通信バンド幅性能を確保しつつ遅延を削減する通信方式が必要となっている。

4. 10Gb Ethernet を用いた高性能通信機構の設計

本章では、第 3 章で述べた課題を解決するための高性能通信機構の設計について述べる。本通信機構の目標は、Pin-down の同期を不要としながら、RDMA 通信と同等の転送バンド幅性能を実現することとする。

Pin-down の同期不要で通信を行うためのアプローチとしては、要求に応じて Pin-down を行うアプローチである動的 Pin-down 方式と、メッセージ通信の手法を用いたアプローチである Buffered Page Replace 方式がある。以下、これらのアプローチについて説明し、機能とコスト比較を行う。RDMA 通信には、Remote Memory Write(RM-Write) と Remote Memory Read(RM-Read) 通信があるが、RM-Read は Remote RM-Write と同等の処理となるため、ここでは、RM-Write 処理について議論する。

4.1 動的 Pin-down 方式

動的 Pin-down 方式は、RDMA 通信の実行時に事前に Pin-down は行わず、転送要求発生時に Pin-down 処理を行なう方式である。

動的 Pin-down 方式においては、仮想アドレス空間を定義する。例えば、ユーザプロセスであればプロセス ID とユーザのアドレス空間、カーネル空間であれば特殊 ID とアドレス空間のように定義する。そして、要求発行者は RDMA 要求の時に、定義済の転送先のアドレスを指定して RDMA 処理を実行する。

要求発行者からの RDMA 処理要求を受けた場合は、転送先アドレスを、ホストプロセッサに対してハードウェア割込みにより通知し、ホストプロセッサで、該当アドレス領域の Pin-down を行い、物理アドレスを NIC に通知した後、ホストメモリとのデータ転送を行う。

本方式は、ハードウェア割込みと Pin-down のオーバヘッドがオリジナルの RDMA 方式に加わる。また、実際に物理メモリが割り当てられていない場合には、物理メモリ割り当てのオーバヘッドが加わる他、該当ページが、ページアウトされていたり割り当てメモリがない場合は処理を中断する。この場合は、再送処理を行う。

また、本方式は、Pin-down cache[4] と併用することにより、再度利用されるエントリについては、実オー

バヘッドを大幅に削減することが可能である。

4.2 Buffered Page Replace 方式

Buffered Page Replace 方式は、メッセージ通信的なアプローチで RDMA に類似した通信を実現する方式である。

Buffered Page Replace 方式においては、予め最大転送長の領域をページ単位で受信バッファとして確保しておく。そして、転送要求が来た場合に、予め割り当てられたページに対してデータ転送を行う。データ転送終了後、本来の転送先のアドレスとページのリストのセットを受信完了を知らせるキューに挿入して、ハードウェア割込みによりホストプロセッサに通知する。ホストの割込みハンドラは、受信完了のキューから該当のアドレスに対して従来 map されていたページを受信完了のキューから取り出したリストのページに置き換える。

この方式の特徴は次の通りである。

- ハードウェア割込みと remap のオーバヘッドがオリジナルの RDMA に加わる。しかし、実際に物理メモリが割り当てられていない場合にも、新しく割り当てるメモリとして map すればよいため、受信用のページがある限り効率的にデータ転送を行うことが可能である。
- ページ全体を置き換える場合には、単にページを置き換えるだけで済む。しかし、ページの全体を置き換えない場合には、境界部分をマージする必要がある。このマージはホストプロセッサによるコピーを行うか、動的 Pin-down 方式を用いるかが可能である。
- ページ全体を置き換える場合で、かつ、既存のページを他のプログラムがアクセスしない場合は、ホストメモリ上へのデータ転送とハードウェア割込みによる map の変更を並行して処理することが出来る。
- メッセージ通信的なアプローチで通信を行うので、受信用のページの供給が必要になる。
- 利用する OS カーネルの仮想メモリ管理との連携が必要である。

4.3 モデルによる処理オーバヘッドの見積もり

本節では、前節で延べた 2 つの方式を従来の RDMA 方式と比較するため RM Write の受信側の処理コストを見積もる。見積には以下の変数を用いる。

$\mathbf{T}_{hwintr}$: ハードウェア割込みのオーバヘッド
$\mathbf{T}_{rtt}$: メッセージのラウンドトリップ時間
$\mathbf{T}_{mlock}$: 1 ページの MLock に要するコスト
$\mathbf{T}_{remap}$: 1 ページの remap に要するコスト
$\mathbf{T}_{pcird}$: 要求キューを読むコスト (16Bytes)
$\mathbf{T}_{pciwr}$: NIC への伝達のコスト (4Bytes)
$\mathbf{N}_{page}$: 転送するページ数
$\mathbf{T}_{pcidma}$: 1 ページあたりの I/O バス転送コスト

本節で用いるモデルは、RM Write 実行時の相手側の処理を扱う。そして、見積もるコストは、I/O バス上のデータ転送を除く方式による追加のオーバヘッドのコストとする。なお、Buffered Page Replace 方式のコスト見積もりはページ全体を置き換える場合について扱う。

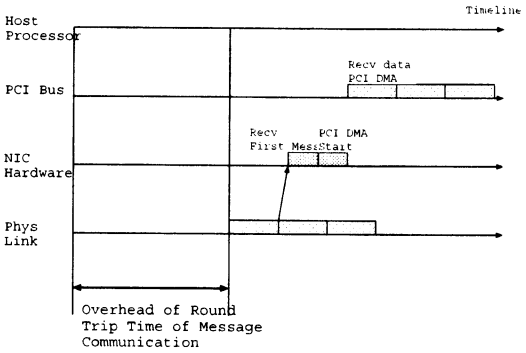

図 2 オリジナル RDMA 通信

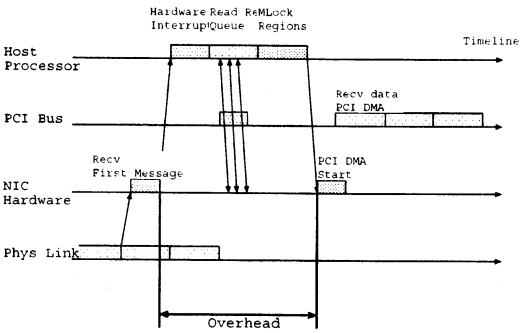

図 3 動的 Pin-down 方式

(1) オリジナル RDMA 通信

図 2 にオリジナル RDMA 通信モデルを示す。このモデルでは、Pin-Down 後にメッセージ交換により同期を行った後に、RM Write 通信を実行する。このモデルのオーバヘッドはメッセージのラウンドトリップ時間と Mlock 時間の和となる。よって、

$T_{rdma} = T_{rtt} + T_{mlock} \times N_{page}$ となる。

(2) 動的 Pin-down 方式

図 3 に動的 Pin-down 方式による通信のモデルを示す。このモデルでは、最初のメッセージ受信時に、RM Write 要求を NIC 上にあるキューに格納した後、ハードウェア割込みによりホストプロセッサに通知する。割込みハンドラにおいて、NIC 上のキューから Pin-down する領域の情報を読み出し、Pin-down 処理を Mlock により実行する。Mlock 処理の完了後に、NIC にその旨を通知し、NIC は Mlock の完了後、DMA 処理によりデータを転送する。本方式で

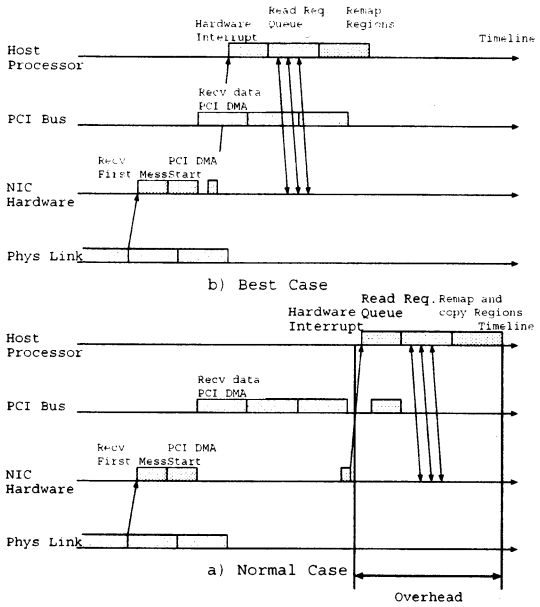

図 4 Buffered Page Replace 方式

は、Pin-down が確定するまで、実際のメモリへのデータ転送を開始できないが、この間に届いたメッセージは NIC 内に格納する必要がある。このモデルのオーバヘッドは、ハードウエア割込みのオーバヘッド、NIC 上のキューを読むオーバヘッド、Pin-down のオーバヘッド、NIC への通知のオーバヘッドの和となる。よって、$T_{dynpin} = T_{hwintr} + T_{mlock} \times N_{page} + T_{pcird} + T_{pciwr}$ となる。

(3) Buffered Page Replace 方式

図 4 に Buffered Page Replace 方式による通信のモデルを示す。このモデルでは、予め蓄えられた page にデータを転送し、そのページを実際の転送先ページと置き換える。通常の処理は、図 4 の a) のように、実際のデータ転送が終了した後に、ハードウェア割込みによりホストプロセッサに通知する。割込みハンドラにおいて、NIC 上のキューから転送先領域の情報と置き換えるページリストを読みだし、該当するページを置き換える。もし、転送先の領域がアクセスされない場合は、図 4 の a) のように、転送先のページのリストが決定した時点で、ハードウェア割込みを上げ、実際のデータ転送と、ハードウェア割込みとページの置き換えは並行して実行できる。

図 4 a) の場合のオーバヘッド： ハードウエア割込みのオーバヘッド、NIC 上のキューを読むオーバヘッド、remap のオーバヘッドの和となる。したがって、

$T_{bufpgrep-a} = T_{hwintr} + T_{remap} \times N_{page} + T_{pcird}$ となる。

図 4 b) の場合のオーバヘッド： $T_{bufpgrep-a}$ の時間が、$T_{pcidma} \times N_{page}$ の時間以下の場合は処理がオーバラップされ見えなくなる。また、$T_{bufpgrep-a}$ の時間が、$T_{pcidma} \times N_{page}$ の時間より大きい場合には、I/O バス転送コストを超えるコストがオーバヘッドとなる。したがって、

- $T_{bufpgrep-b} = 0$ (ただし、$T_{bufpgrep-a} \leq T_{pcidma} \times N_{page}$ の場合)
- $T_{bufpgrep-b} = T_{bufpgrep-a} - T_{pcidma} \times N_{page}$ (ただし、$T_{bufpgrep-a} > T_{pcidma} \times N_{page}$ の場合)

となる。

以上延べたオーバヘッドの計算式を、表 1 にまとめる。表 1 より、適用が可能であれば、明らかに Buffered Page Replace 方式の b) Best Case が優れている。それ以外では、実際の各コストと転送ページ数により、優劣が決る。

表 1 各方式のコスト式

方式	コスト式
RDMA	$T_{rdma} = T_{rtt} + T_{mlock} \times N_{page}$
動的 Pin-down	$T_{dynpin} = T_{hwintr} + T_{mlock} \times N_{page} + T_{pcird} + T_{pciwr}$
Buffered Page Replace a)	$T_{bufpgrep-a} = T_{hwintr} + T_{remap} \times N_{page} + T_{pcird}$
Buffered Page Replace b)	$T_{bufpgrep-b} = T_{bufpgrep-a} - T_{pcidma} \times N_{page}$ (ただし、$T_{bufpgrep-a} > T_{pcidma} \times N_{page}$ の場合) $T_{bufpgrep-b} = 0$ (ただし、$T_{bufpgrep-a} \leq T_{pcidma} \times N_{page}$ の場合)

4.4 実測データによる処理コスト比較

本節では表 1 にあげた各方式に基づき各コストの実測値を代入し、方式の比較を行う。測定のハードウェア環境として、Xeon 2.8GHz、ServerWorks GC LE チップセット (FSB 533MHz) を搭載した PC を採用した。OS は Linux 2.4.21 である。ページサイズは、4KB である。実測は、CPU の持つカウンタを用いて行った。

表 2 に各種コストの測定結果、表 3 に表 2 の各種コストを代入したコスト式を示す。表 3 より、T_{hwintr} のコストが大きいことが分かる。

図 5 に、表 3 の各方式のコスト式をページ数を可変にした場合の計算結果を示す。またページ数が 1,64,256 の時に置ける各方式のオーバヘッドを表 4 に示す。

オリジナルの RDMA 方式の T_{rtt} については、ネットワークの純粋なラウンドトリップタイムを $T_{rtt-net}$

表 2　各種コスト測定結果

	Cost		Cost
T_{hwintr}	6.5 μsec	T_{mlock}	0.014 μsec
T_{remap}	0.064 μsec	T_{pcird}	1.9 μsec
T_{pciwr}	0.1 μsec	T_{pcidma}	3.9 μsec

表 3　各方式のコスト式

方式	コスト式
RDMA	$T_{rdma} = T_{rtt} + 0.014 \times N_{page}$
動的 Pin-down	$T_{dynpin} = 6.5 + 0.014 \times N_{page} + 1.9 + 0.1$
Buffered Page Replace a)	$T_{bufpgrep-a} = 6.5 + 0.064 \times N_{page} + 1.9$
Buffered Page Replace b)	$T_{bufpgrep-b} = T_{bufpgrep-a} - 3.9 \times N_{page}$ (ただし、$T_{bufpgrep-a} > T_{pcidma} \times N_{page}$ の場合) $T_{bufpgrep-b} = 0$ (ただし、$T_{bufpgrep-a} \leq T_{pcidma} \times N_{page}$ の場合)

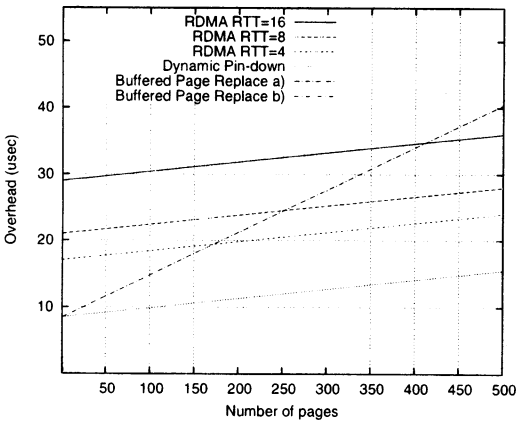

図 5　各方式のオーバヘッド

= 4, 8, 16μsec とした場合の結果を示している。

第 2 章で述べたように、CPU 占有率を抑えるため、ハードウェア割込みを用いた通信を用いるとすると、$T_{rtt} = T_{rtt-net} + 2 \times T_{hwintr}$ となる。

表 4　ページ数が 1,16,256 の時のオーバヘッド

	1	16	256
RDMA RTT=16	29.01	29.22	32.58 μsec
RDMA RTT=8	21.01	21.22	24.58 μsec
RDMA RTT=4	17.01	17.22	20.58 μsec
動的 Pin-down	8.51	8.72	12.08 μsec
B.P.R a)	8.46	9.24	24.78 μsec
B.P.R b)	4.56	0.00	0.00 μsec

B.P.R: Bufferd Page Replace 方式

図 5 において、Buffered Page Replate b) 方式の計算結果は、ページ数が 1 の場合のオーバヘッドは 4.5μsec、ページ数が 2 を超える場合から 0 となっており、理想的な結果となっている。

また、オリジナルの RDMA 方式は、利用するネットワークのラウンドトリップタイムに大きく影響されることがわかる。

5. 議　　論

本章では、第 4 章で議論した動的 Pin-down 方式、Buffered Page Replace 方式、オリジナル RDMA 方式と比較議論する。論点は、適用性、オーバヘッド、移植性である。

5.1 適 用 性

オリジナル RDMA 方式と、動的 Pin-down 方式適応性は同じである。Buffered Page Replace 方式は、ページ単位での置き換えが基本になるため他の方式に比べ制限がある。しかし、第 2 章で述べた要件に照らすと、適用範囲は広いと考えられる。例えば、Buffered Page Replace 方式 b) は、Swap ページへの書き出しなど、転送先領域のオリジナルデータを外のプログラムが、参照しない場合に適用可能と考えている。

5.2 オーバヘッド

第 4.4 節の議論より、Buffered Page Replace 方式 b) が最もオーバヘッドが小さいと言える。次にオーバヘッドが小さいのは動的 Pin-down 方式である。また、オリジナル RDMA はハードウエア割込みが 1 回の転送に 2 回必要になるため、他の 2 つの方式に比べ、オーバヘッドが大きい。

5.3 CPU 占有率

第 4.4 節の議論より、動的 Pin-down 方式と Buffered Page Replace 方式はハードウェア割込みの回数がオリジナル RDMA 方式の 2 回に比べて 1 回で済むため、CPU 占有率が少ないと考えられる。

5.4 移 植 性

割込みハンドラ内での処理がシンプルであるため、オリジナル RDMA 方式がもっとも移植性が高い。動的 Pin-down 方式は、割込みハンドラ内での Pin-down 処理が必要、Buffered Page Replace 方式は、OS の仮想記憶との連携が必要であるため、移植性はオリジナル RDMA 方式に比べて良くない。

6. 関 連 研 究

RDMA 通信を実現した高性能通信機構は、Myrinet を [5] 用いた GM[6], BIP[7], VMMC-2[8], PM[9], MX[10], InfiniBand[11] があるが、これらは、本論文で述べているオリジナル RDMA 通信を採用している。オリジナル RDMA 通信は、Pin-down の領域をユーザプログラムもしくは、オペレーティングシステムで適切に管理して、全体メモリ量に対する Pin-down を実施するメモリ量を適切に制御して、OS の動

作に悪影響を与えないようにする必要がある。また、Pin-down されているかの保証のため、ネットワークを用いた同期をとる必要がある。

本論文で対象にしている通信方式は、以上の問題を回避する方式である。

7. おわりに

本論文では、10Gbps ネットワークを用いた高信頼かつ高性能システム向けの通信機構の設計について述べた。現状のハードウェアで 10Gbps の通信性能を生かし切るには、RDMA 通信を用いる必要がある。しかし、現状の RDMA 通信モデルは Pin-down する領域を適切に管理するために、ネットワークを用いた同期が必要な点が問題である。本論文では、この問題を解決するための通信方式を議論し、動的 Pin-down 方式と Buffered Page Replace 方式を提案し、モデルを用いてオーバヘッドを見積もったところ、従来の RDMA 方式に比べオーバヘッドを減らせることがわかった。

今後は、実システムに適用した場合の効果についてより深く検討を進め、実装評価する予定である。

謝辞 本研究の一部は、文部科学省「eSociety 基盤ソフトウェアの総合開発」の委託を受けた東京大学石川研究室および東京大学石川研究室と富士通研究所との共同研究契約に基づいて行なわれた。

参考文献

1) 石川裕, 住元真司, 岡家豊, 久門耕一, 木村かず子. 次世代高性能計算機アーキテクチャ上のシステムソフトウェア開発環境. 情報処理学会研究報告 03-OS-94 (SWoPP'2003). 情報処理学会, August 2003.
2) Shinji SUMIMOTO, Hiroshi TEZUKA, Atsushi HORI, Hiroshi HARADA, Toshiyuki TAKAHASHI, and Yutaka ISHIKAWA. High Performance Communication using a Commodity Network for Cluster Systems. In *the Ninth International Symposium on High Performance Distributed Computing (HPDC-9)*, pp. 139–146. IEEE, August 2000.
3) Shinji Sumimoto and Kouichi Kumon. PM/Ehernet-kRMA: A High Performance Remote Memory Access Facility Using Multiple Gigabit Ethernet Cards. In *3rd International Symposium on Cluster Computing and the Grid*, pp. 326–334. IEEE, May 2003.
4) Hiroshi Tezuka, Francis O'Carroll, Atsushi Hori, and Yutaka Ishikawa. Pin-down Cache: A Virtual Memory Management Technique for Zero-copy Communication. In *IPPS/SPDP'98*, pp. 308–314. IEEE, April 1998.
5) N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic and Wen-King Su. Myrinet - a gigabit-per-second local-area network. *IEEE MICRO*, Vol. 15, No. 1, pp. 29–36, February 1995.
6) The GM API: http://www.myri.com/GM/doc/gm_toc.html.
7) L. Prylli and B. Tourancheau. BIP: a new protocol designed for high performance. In PC-NOW Workshop, held in parallel with IPPS/SPDP98, Orlando, USA, Mar 30 – Apr 3 1998.
8) C. Dubnicki, A. Bilas, Y. Chen, S. Damianakis, and K. Li. VMMC-2: Efficient Support for Reliable, Connection-Oriented Communication. In Hot Interconnect'97, August 1997.
9) Hiroshi Tezuka, Atsushi Hori, Yutaka Ishikawa, and Mitsuhisa Sato. PM: An Operating System Coordinated High Performance Communication Library. In Peter Sloot Bob Hertzberger, editor, *High-Performance Computing and Networking*, Vol. 1225 of *Lecture Notes in Computer Science*, pp. 708–717. Springer-Verlag, April 1997.
10) Myricom MX: http://www.myri.com/scs/MX/doc/mx.pdf.
11) InfiniBand Trade Association: http://www.infinibandta.org/.